ONLINE HANDWRITTEN GUJARATI CHARACTER RECOGNITION USING SVM, MLP, AND K-NN

Vishal A. Naik and Apurva A. Desai
Department of Computer Science
Veer Narmad South Gujarat University
Surat, Gujarat, India
vishalkumar.naik@gmail.com, aadesai@vnsgu.ac.in

Abstract:

In this paper, we present a system to recognize online handwritten character for the Gujarati language. Support Vector Machine (SVM) with linear, polynomial & RBF kernel, k-Nearest Neighbor (k-NN) with different values of k and multi-layer perceptron (MLP) are used to classify strokes using hybrid feature set. This system is trained using a dataset of 3000 samples and tested by 100 different writers. We have achieved highest accuracy of 91.63% with SVM-RBF kernel and lowest accuracy of 86.72% with MLP. We have achieved minimum average processing time of 0.056 seconds per stroke with SVM linear kernel and maximum average processing time of 1.062 seconds per stroke with MLP.

Keywords: Online Handwritten Character Recognition (OHCR), Handwritten Character Recognition (HCR), Optical Character Recognition (OCR), Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), multi-layer perceptron (MLP), Gujarati character Recognition

1. Introduction:

The input to handheld devices using a traditional keyboard is not a user-friendly process for Indian scripts due to large and complex character sets. Handwritten character recognition can be a best possible solution. Handwritten character recognition is gaining noteworthy attention in the area of pattern matching and machine learning. Handwritten character recognition can be classified into online and offline based on the mode of data acquisition. In offline character recognition, an image of a handwritten character is captured using a scanner, features are extracted and recognized using a feature set. In Online character recognition, it captures the pixel values based on the movement of a cursor or a stylus and converts it into text using a classifier and feature set.

The Gujarati is a native and official language of Indian state Gujarat, which is an Indo-Aryan type language. Gujarati has many similarities with Devanagari language. The Gujarati can be easily differentiated from any other Indic language because it does not have Shirolekha over its characters. The Gujarati script character set consists of numerals, consonants, and vowels. Figure 1 illustrates Gujarati Characters. For some characters, multiple strokes are required to make a single character. For example, character 'ઝ' requires 3 different strokes. Some strokes are used in multiple characters. In figure 1, colored strokes indicate repeating stroke. Many characters have high resemblance with other characters, like 'ધ' and 'ઘ'.

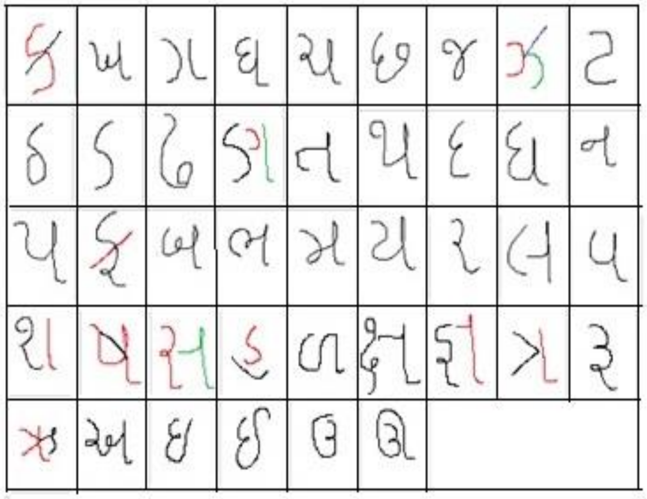


Figure.1 Gujarati Characters

Many researchers are working in the area of handwritten character recognition for different Indian languages.

In [1] the authors has presented work on segmentation of characters from old typewritten Gujarati documents. He has used different preprocessing techniques and used radon transform for a line, word and character segmentation.

In [2] the authors have presented work on segmentation of text line into words for Gujarati handwritten text. They have proposed morphological operation and projection profile based algorithm and achieved 89.24% accuracy.

In [3] the authors have presented work on zone identification of Gujarati handwritten word. They have proposed Euclidean transform based zone identification and achieved 83.6% accuracy and achieved 96.99% accuracy.

In [4] the author has presented work on handwritten Gujarati numeral using hybrid features like a sub division of skeletonized image and aspect ratio as a statistical approach and used k-NN classifier with Euclidean distance method and achieved 82% accuracy.

In [5] the author proposed similar work using four profile vector based feature set and used multilayer feed forward neural network and achieved 82% accuracy.

In [6] the author proposed similar work using hybrid feature set that includes aspect ratio, extent, and 16 other features by dividing an image into 4x4 sub-images. He has used SVM with a polynomial kernel for classification and achieved 86.66% accuracy.

In [7] the authors have used k-Nearest Neighbors using low level and directional features for Gujarati characters and achieved 95% accuracy.

In [8] the authors have used K-Nearest Neighbor, Support Vector Machine and Back Propagation Neural Network using spatial and transform domain features for Gujarati numerals and achieved 93.6% accuracy.

In [9] the authors have combined HMM and SVM using resampled coordinate sequence, 1st and 2nd order derivatives feature for Assamese characters and achieved 96.17% accuracy.

In [10] the authors have used two-stage classification approach using HMM and SVM using frequent count, used first and second order derivatives, slope, and base line features for Assamese and achieved 95.1% accuracy.

In [11] the authors have used weighted Euclidean distance for classification using 48 geometric features for Odia and achieved 87.6% accuracy.

In [12] the authors have used different geometric values as fuzzy features for recognition of Bangla character and achieved 77% accuracy.

In [13] the authors have used FFNN using hybrid features of structural features, zoning and optimally fitted curve for Devanagari and achieved 93.4% accuracy.

In [14] the authors have used HMM and Bayesian classifier using feature set of zoning, directional, diagonal, intersections, and Zernike moments individually and an average of all features for Gurumukhi and achieved 93.5% accuracy.

In [15] the authors have used SVM, k-NN and FFNN using hybrid features set consist of geometric, regional, gradient and distance transform features for Devanagari and achieved 91.3% accuracy.

In [16] the authors have used ANN using structural and wavelet transform features for Kannada and achieved 91% accuracy.

In [17] the authors have used SVM using different global and local features for Arabic characters and achieved 92.43% accuracy.

In [18] the authors have used a deep convolutional neural network (CNN) to identify similar Chinese characters and achieved 98.44% accuracy.

The rest of the paper is organized as follows. Section 2 describes pre-processing methods. Section 3 describes feature extraction methods. Section 4 describes classifiers. Section 5 describes results and discussion. Section 6 describe the conclusion of the paper.

2. Pre-Processing:

The written stroke of a character (a sequence of (x, y) coordinates) is pre-processed before extracting features from it. Following pre-processing methods are used here, normalization, Smoothing and Resampling. Height and width of stroke should be normalized to remove variation in size among all strokes. Bilinear interpolation method is used to normalize a stroke. The bilinear interpolation method will perform linear interpolation first in one direction and then in the other direction. Smoothing is performed to remove noise from a stroke. The median filter is a nonlinear filtering technique used for smoothing. Resampling is used to remove the effect of variation in speed of writing stroke, if speed is low then we will get more coordinates and if speed is high then we will get less coordinates. Pixel area based image decimation resampling method is used here to resample the recorded stroke coordinates.

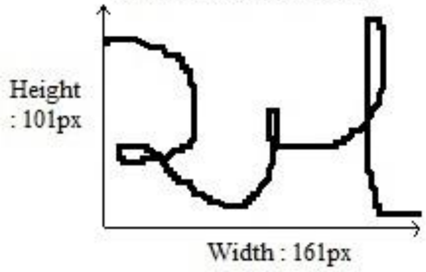
3. Feature extraction:

Pre-processed coordinates contain plenty of raw data. We have to use different methods to extract different types of unique meaning full information from raw data. Each and every stroke contains some unique information, known as a feature. Different types of unique features of stroke are extracted using different methods. Extracted features will be used to classify the characters.

Different structural, statistical and hybrid feature extraction methods can be used to extract different global and local features. We have used hybrid feature set which is extracted using different structural and statistical feature extraction methods. Proposed feature set includes aspect ratio, zoning, start & end zone and normalized chain code features which are a size and speed independent.

Aspect ratio is a global feature which describes a relationship between width and height of the stroke, written as x: y where x denote width and y denotes height unit. Figure 1 illustrates aspect ratio of character “અ”.

Figure 1. Aspect ratio 8:5



Zoning is a local feature which describes percentage wise active pixel distribution in different zones of a stroke. Stroke is divided into 16 equal zones by considering its length and width. Figure 2 illustrates zoning of character “અ” with a percentage of active pixels in each zone. Zone of starting and ending of stroke are also considered as features. Starting and ending zone of character “અ” is Zone 1 and zone 16.

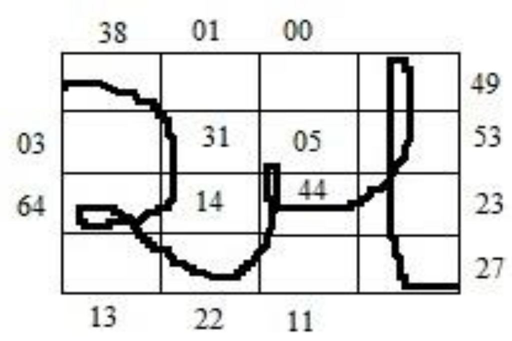


Figure 2. Character “અ” in 16 equal zones with percentage of active pixels

Normalized chain code describes stroke's curvature using directional information. Stroke’s direction is measured after every 10% of total active pixels. Figure 3 illustrates different chain code values for a different direction. Figure 4 illustrates stroke’s direction with chain code value for character “અ”. The extracted chain code values for character “અ” is 7,6,5,7,1,0,1,3,6,7.

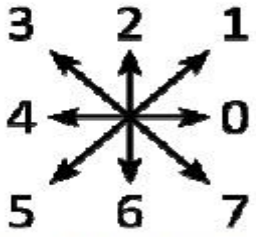


Figure 3. Chain Code for each Direction

We can normalize a chain code to make it starting point and rotation invariant. Consider the chain code as a circular and redefine the starting point so that the resulting chain code forms an integer of minimum magnitude. The starting point normalized chain code values for character “અ” is 0,1,3,6,7,7,6,5,7,1.

To make chain code values rotation normalized, use the first difference of the chain code. Calculating difference code is simply by counting the number of directions that separate two adjacent elements of the code. Counting of direction is counter-clockwise. The rotation normalized chain code values for character “અ” is 1,2,3,1,0,7,7,2,2,7.

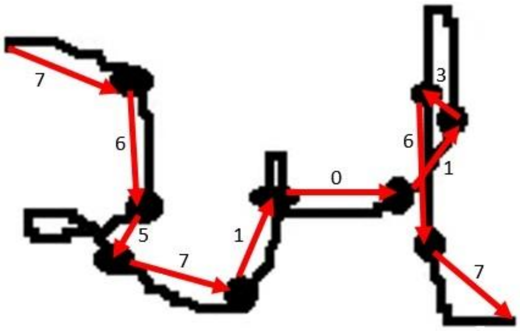


Figure 4. Directional chain code for character “અ”

4. Classification:

We have tested k-NN, SVM and ANN classification methods to classify characters.

K-Nearest Neighbor Classifier (k-NN) is a distance-based classification method. It computes a Euclidean distance between testing data with all data samples of a training set. K samples having minimum distance with testing sample will be selected from the training set. The class label for a majority of k-nearest data samples from training set will be set for a testing sample. Different k values lead to different processing and results. A smaller value may lead to low accuracy and higher value may lead to more processing. To choose an optimum value of k, we have started with a thumb rule of a square root of classes. We have tested three different values of k, k=5, k=7 and k=9.

Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for classification of one or more classes. SVM uses different kernels which can effectively compute dot products to implicitly transform data into higher dimension. These kernels require no extra memory and minimum effect on processing time. With the help of kernels, SVM can gain flexibility in selection of the threshold. SVM will find global optimum and unique solution due to quadratic programming and it is also less prone to overfitting. We have used three different kernels like polynomial, radial basis function and linear.

Polynomial kernel is defined as, $K (X_{i,} X_j) = (\gamma X_i^T X_j + \text{Coef0})^{\text{degree}}$, $\gamma > 0$. We have used constant values as, γ =0.33, Coef0 = 0, degree=1 and C = 1.

Radial basis function kernel is defined as, $K (X_{i,} X_j) = {}_{e} - \gamma \| x_i – x_j \|^2$, $\gamma > 0$. We have used constant values as, γ= 0.002 and C =1.

A multi-layer perceptron (MLP) is a feed forward type of artificial neural network. A MLP has multiple layers. Each layer has some nodes. Each layer is connected to the next layer. Each node is known as a neuron. Each node has a nonlinear activation function except input nodes. We have used sigmoid activation function here. We have used 28 input neurons, one hidden layer with 10 neurons and 1 neuron in output layer. We have used a back propagation technique to train the network.

5. Results and discussion:

The proposed system has training dataset of around 3000 samples collected from different writers of different age group and gender. The proposed system is tested by 100 different writers with around 55 samples each. All the training data and testing data captured using developed GUI system.

We have achieved highest accuracy of 91.63% using SVM with RBF kernel and lowest accuracy of 86.72% using MLP. SVM with poly kernel took minimum average processing time of 0.056 seconds and maximum average processing time of 1.062 using MLP. Table 1 shows a comparison between SVM, k-NN and MLP classifiers in terms of accuracy and average processing time.

Classifier	No. of Successful samples	Accuracy rate (%)	Avg. Processing Time (Seconds)
SVM (Poly kernel)	5015	91.18	0.072
SVM (Linear kernel)	4985	90.63	0.056
SVM (RBF kernel)	5040	91.63	0.063
MLP	4770	86.72	1.062
k-NN (k=5)	4940	89.81	0.156
k-NN (k=7)	4955	90.09	0.165
k-NN (k=9)	4935	89.72	0.171

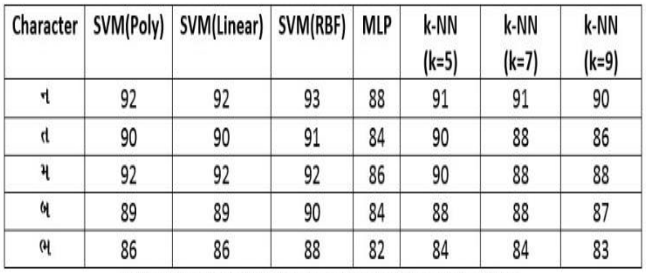
Table 1 Comparison between classifiers

Table 2 describes the accuracy of high resemblance characters in different classifiers. These characters’ have similarity with other characters in some part of the stroke. These confusing characters lead to misclassification.

Character	SVM(Poly)	SVM(Linear)	SVM(RBF)	MLP	k-NN (k=5)	k-NN (k=7)	k-NN (k=9)
ધ	86	85	84	80	85	85	84
દ	86	84	86	82	86	86	86
થ	92	92	93	89	91	91	90
ચ	91	91	94	88	90	91	90
પ	90	90	92	84	88	88	86
ય	84	83	88	82	86	86	84
વ	90	88	90	84	90	90	88

Table 2 Accuracy of high resemblance characters in different classifiers

Another set of confusing characters' accuracy is described in table 3. These characters have fewer similarities compare to above mentioned characters.



Character	SVM(Poly)	SVM(Linear)	SVM(RBF)	MLP	k-NN (k=5)	k-NN (k=7)	k-NN (k=9)
ન	92	92	93	88	91	91	90
ત	90	90	91	84	90	88	86
મ	92	92	92	86	90	88	88
અ	89	89	90	84	88	88	87
ભ	86	86	88	82	84	84	83

Table 3 Accuracy of similar characters in different classifiers

There are many characters which require multiple strokes to draw a single character. To recognize such characters, we have used simple association rules. Sequence based association rules are defined to recognize such characters. If a predefined set of strokes appear in a predefined sequence then only that character is recognized. Figure 5 illustrates such character 'ક' which requires two different strokes.

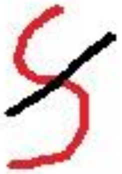
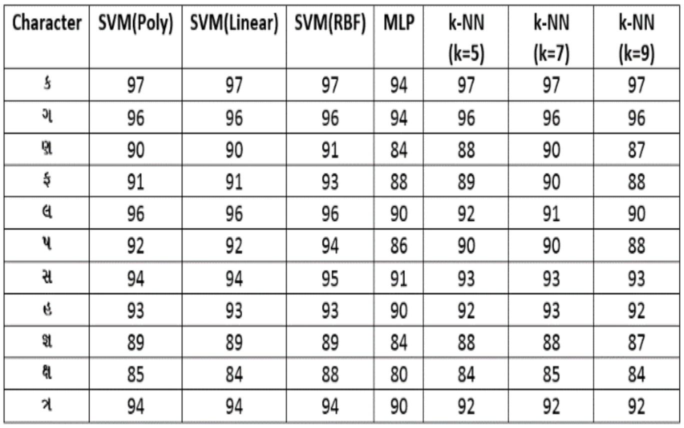


Figure 5. Multi stroke Character 'ક'

There are 11 such multi-stroke characters. Table 4 describes the accuracy of different multi-stroke characters in different classifiers.



Character	SVM(Poly)	SVM(Linear)	SVM(RBF)	MLP	k-NN (k=5)	k-NN (k=7)	k-NN (k=9)
ક	97	97	97	94	97	97	97
ગ	96	96	96	94	96	96	96
ણ	90	90	91	84	88	90	87
ફ	91	91	93	88	89	90	88
લ	96	96	96	90	92	91	90
પ	92	92	94	86	90	90	88
સ	94	94	95	91	93	93	93
હ	93	93	93	90	92	93	92
શ	89	89	89	84	88	88	87
ક્ષ	85	84	88	80	84	85	84
ત્ર	94	94	94	90	92	92	92

Table 4 Accuracy of multi stroke characters in different classifiers

6. Conclusion:

We proposed an algorithm for online handwritten Gujarati character recognition using hybrid features. We have compared SVM, MLP and k-NN classifiers using different parameter values. We have used a hybrid feature set with a training set of around 3000 samples. The proposed system is tested by 100 different users. We have achieved highest accuracy of 91.63% and 0.063 seconds of average execution time per stroke using SVM with RBF kernel. SVM with linear kernel executed fastest with 0.056 seconds of average execution time per stroke and 90.63% accuracy. The limitation of proposed system is that it provides low accuracy for high resemblance, similar characters and confusing characters. A two layer classifier approach can be used to improve recognition rate for confusing characters.

Acknowledgments: the authors acknowledges the support of University Grants Commission (UGC), New Delhi, for this research work through project file no. F. 42-127/2013.

References:

1 Apurva Desai, "Segmentation of Characters from old Typewritten Documents using Radon Transform", International journal of Computer Application, Vol.37-No.9, pg.10-15 (2012)

2 Chhaya Patel, Apurva Desai, "Segmentation of text lines into words for Gujarati handwritten text", International conference on Signal and Image Processing, pg. 130-134 (2010)

3 Chhaya Patel, Apurva Desai, "Zone Identification for Gujarati Handwritten Word", Second international conference on Emerging application of Information technology, pg. 194-197 (2011)

4 Apurva A. Desai, "Handwritten Gujarati Numeral Optical Character Recognition using Hybrid Feature Extraction Technique", Image Processing, Computer Vision, & Pattern Recognition (IPCV) (2010)

5 Apurva A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network", Pattern Recognition, Vol.43 Issue 7, pp. 2582-2589 (2010)

6 Apurva A. Desai, "Support vector machine for identification of handwritten Gujarati alphabets using hybrid feature space", CSI Transactions on ICT by Springer (2015)

7 C. Gohel, M. Goswami, V. Prajapati, "On-line Handwritten Gujarati Character Recognition Using Low Level Stroke", Third International Conference on Image Information Processing, pp 130-134 (2015)

8 Archana Vyas and Mukesh Goswami, "Classification of handwritten Gujarati numerals", International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp 1231-1237 (2015)

9 H. Choudhury, S. Mandal, S. Devnath, S. R. M. Prasanna and S. Sundaram, "Combining HMM and SVM based stroke classifiers for online Assamese handwritten character recognition", Annual IEEE India Conference (INDICON) (2015)

10 S. Mandal, H. Choudhury, S. R. M. Prasanna and S. Sundaram, "Frequency Count based Two Stage Classification for Online Handwritten Character Recognition", International Conference on Signal Processing and Communications (SPCOM) (2016)

11 I. Rushiraj, S. Kundu, B. Ray, "Handwritten Character Recognition of Odia Script",International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)(2016)

12 K. Chowdhury, L. Alam, S. Sarmin, S. Arefin, and M. Mo. Hoque, "A Fuzzy Features Based Online Handwritten Bangla Word Recognition Framework", 18th International Conference on Computer and Information Technology (ICCIT), 2015 , pp 484-489

13 D. KHANDUJA, N. NAIN, and S. PANWAR, "A Hybrid Feature Extraction Algorithm for Devanagari Script", ACM Transactions on Asian and Low-Resource Language Information Processing, volume 15 Issue 1, January 2016

14 Munish Kumar, M. K. Jindal, R. K. Sharma, "A Novel Framework for Grading of Writers Using Offline Gurmukhi Characters",National Academy of Sciences, India Section A: Physical Sciences, September 2016, Volume 86, Issue 3, pp 405–415

15 Saniya Ansari and Udaysingh Sutar, "Devanagari Handwritten Character Recognition using Hybrid Features Extraction and Feed Forward Neural Network Classifier", International Journal of Computer Applications, Volume 129 – No.7, November2015, pp 22-27

16 Saleem Pasha and M.C.Padma, "Handwritten Kannada Character Recognition using Wavelet Transform and Structural Features", International Conference on Emerging Research in Electronics, Computer Science and Technology, 2015, pp 346-351

17 H. Nakkach, S. Hichri, S. Haboubi and H. Amiri, "Hybrid Approach to Features Extraction for Online Arabic Character Recognition", 13th International Conference Computer Graphics, Imaging and Visualization, pp 253-258 (2016)

18 Shuye Zhang, Lianwen Jin, Liang Lin, "Discovering similar Chinese characters in online handwriting with deep convolutional neural networks", International Journal on Document Analysis and Recognition (IJDAR),September 2016, Volume 19, Issue 3, pp 237–252